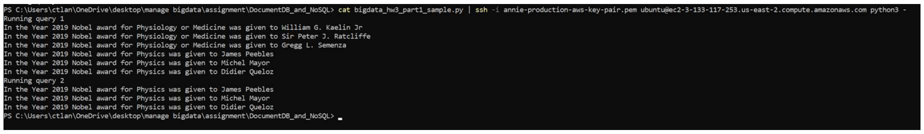
Contents

1 Part1

1.1 Query result of part1



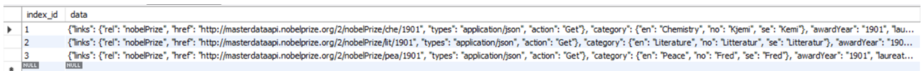
```
PS C:\Users\ctlan\OneDrive\desktop\manage bigdata\assignment\DocumentDB_and_NoSQL> cat bigdata_hw3_part1_sample.py | ssh -i annie-production-aws-key-pair.pem ubuntu@ec2-3-133-117-253.us-east-2.compute.amazonaws.com python3 -
Running query 1
In the Year 2019 Nobel award for Physiology or Medicine was given to William G. Kaelin Jr
In the Year 2019 Nobel award for Physiology or Medicine was given to Sir Peter J. Ratcliffe
In the Year 2019 Nobel award for Physiology or Medicine was given to Gregg L. Semenza
In the Year 2019 Nobel award for Physics was given to James Peebles
In the Year 2019 Nobel award for Physics was given to Michel Mayor
In the Year 2019 Nobel award for Physics was given to Didier Queloz
Running query 2
In the Year 2019 Nobel award for Physics was given to James Peebles
In the Year 2019 Nobel award for Physics was given to Michel Mayor
In the Year 2019 Nobel award for Physics was given to Didier Queloz
PS C:\Users\ctlan\OneDrive\desktop\manage bigdata\assignment\DocumentDB_and_NoSQL>
```

Figure 1: Query result of part1

2 Part2

2.1 Table1: Key Value

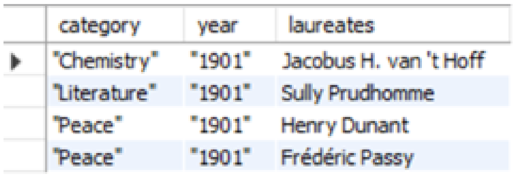
2.1.1 Table1 SnapShot



index_id	data
1	{"links": {"rel": "nobelPrize", "href": "http://masterdataapi.nobelprize.org/2/nobelPrize/che/1901", "types": "application/json", "action": "Get"}, "category": {"en": "Chemistry", "no": "Kjemi", "se": "Kemi"}, "awardYear": "1901", "lau...
2	{"links": {"rel": "nobelPrize", "href": "http://masterdataapi.nobelprize.org/2/nobelPrize/lit/1901", "types": "application/json", "action": "Get"}, "category": {"en": "Literature", "no": "Litteratur", "se": "Litteratur"}, "awardYear": "190...
3	{"links": {"rel": "nobelPrize", "href": "http://masterdataapi.nobelprize.org/2/nobelPrize/pea/1901", "types": "application/json", "action": "Get"}, "category": {"en": "Peace", "no": "Fred", "se": "Fred"}, "awardYear": "1901", "laureat...
NULL	NULL

Figure 2: Table1 SnapShot

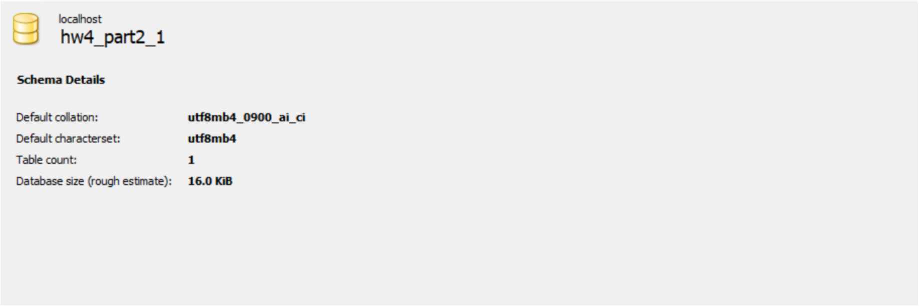
2.1.2 Table1 Query Result



category	year	laureates
"Chemistry"	"1901"	Jacobus H. van 't Hoff
"Literature"	"1901"	Sully Prudhomme
"Peace"	"1901"	Henry Dunant
"Peace"	"1901"	Frédéric Passy

Figure 3: Table1 Query Result

2.1.3 Table1 Table Size



localhost
hw4_part2_1

Schema Details

Default collation:	utf8mb4_0900_ai_ci
Default characterset:	utf8mb4
Table count:	1
Database size (rough estimate):	16.0 KiB

Figure 4: Table1 Table Size

2.1.4 Table1 Load Data Code – python

```
literate
import json
import pymysql
```

```
# Read JSON file
with open(r'C:\Users\ctlan\OneDrive\desktop\manage␣
    bigdata\assignment\DocumentDB␣and␣NoSQL\json_award.json',
    'r') as f:
    data = json.load(f)

# Connect to MySQL
connection = pymysql.connect(host='localhost',
                             user='root',
                             password='@@Ct123456',
                             database='hw4_part2_1',
                             charset='utf8mb4')

cursor = connection.cursor()

# Create table if it doesn't exist
sql_create_table = """
CREATE TABLE IF NOT EXISTS nobel_key_valu (
    index_id INT PRIMARY KEY AUTO_INCREMENT,
    data JSON NOT NULL
);
"""
cursor.execute(sql_create_table)
connection.commit()

# Insert JSON data into table
for i, item in enumerate(data):
    if i >= 3:  # Stop after inserting 3 rows
        break
    sql = "INSERT␣INTO␣nobel_key_valu␣(data)␣VALUES␣(%s);"
    cursor.execute(sql, (json.dumps(item),))

# Commit changes and close connection
connection.commit()
connection.commit()
cursor.close()
connection.close()
```

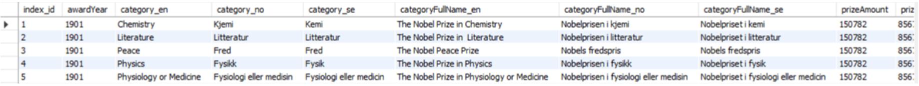
2.1.5 Table1 Query Code – sql

```
literate
use hw4_part2_1;
select
        category,
    year,
    laureates
from
(
        select
                json_extract(data,’$.category.en’) category,
                json_extract(data,’$.awardYear’) year,
                json_names.names laureates
                from nobel_key_valu,
                JSON_TABLE(json_extract(data,
    ’$.laureates[*].knownName.en’)
                , ’$[*]’ COLUMNS (names VARCHAR(255) PATH ’$’))
    AS json_names
                where json_extract(data,’$.awardYear’) = ’1901’
) a
group by
        category,
    year,
    laureates
;

use hw4_part2_2;
select
-- awardYear,
-- category_en,
-- laureate_name_en
*
from nobel_document
where awardYear =’1901’;
```

2.2 Table2: Document

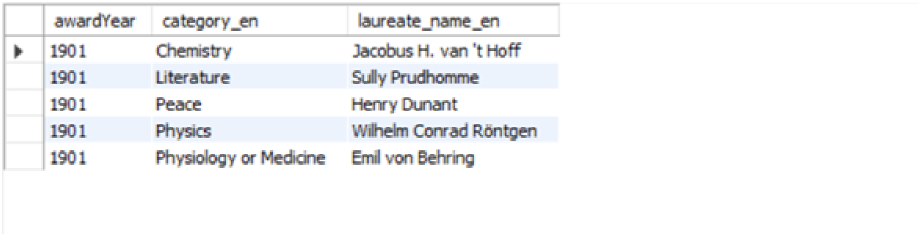
2.2.1 Table2 SnapShot



	index_id	awardYear	category_en	category_no	category_se	categoryFullName_en	categoryFullName_no	categoryFullName_se	prizeAmount	priz
▸	1	1901	Chemistry	Kjemi	Kemi	The Nobel Prize in Chemistry	Nobelprisen i kjemi	Nobelpriset i kemi	150782	856
	2	1901	Literature	Litteratur	Litteratur	The Nobel Prize in Literature	Nobelprisen i litteratur	Nobelpriset i litteratur	150782	856
	3	1901	Peace	Fred	Fred	The Nobel Peace Prize	Nobels fredspris	Nobels fredspris	150782	856
	4	1901	Physics	Fysikk	Fysik	The Nobel Prize in Physics	Nobelprisen i fysikk	Nobelpriset i fysik	150782	856
	5	1901	Physiology or Medicine	Fysiologi eller medisin	Fysiologi eller medicin	The Nobel Prize in Physiology or Medicine	Nobelprisen i fysiologi eller medisin	Nobelpriset i fysiologi eller medicin	150782	856

Figure 5: Table2 SnapShot

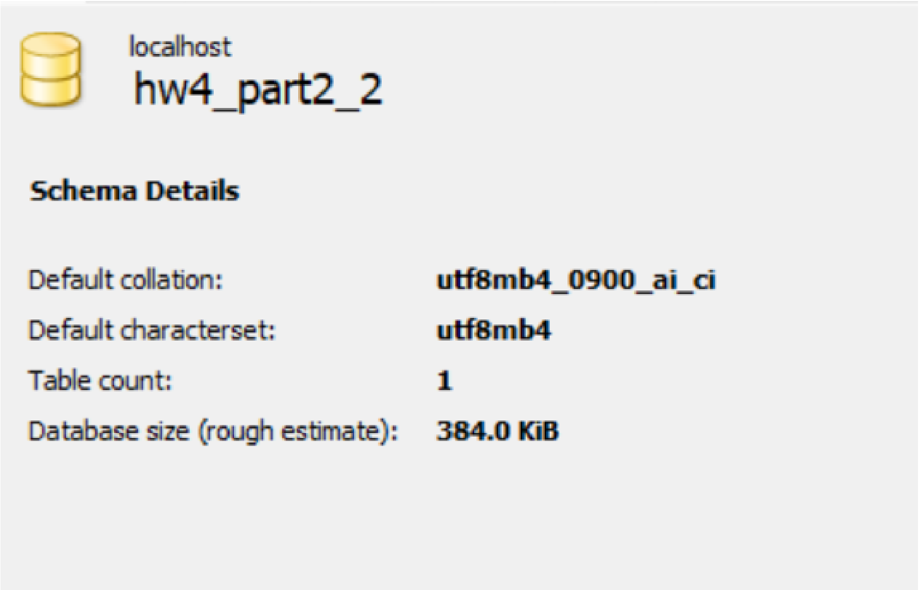
2.2.2 Table2 Query Result



awardYear	category_en	laureate_name_en
1901	Chemistry	Jacobus H. van 't Hoff
1901	Literature	Sully Prudhomme
1901	Peace	Henry Dunant
1901	Physics	Wilhelm Conrad Röntgen
1901	Physiology or Medicine	Emil von Behring

Figure 6: Table2 Query Result

2.2.3 Table2 Table Size



localhost

hw4_part2_2

Schema Details

Default collation:	utf8mb4_0900_ai_ci
Default characterset:	utf8mb4
Table count:	1
Database size (rough estimate):	384.0 KiB

Figure 7: Table2 Table Size

2.2.4 Table2 Load Data Code – python

```
literate
import json
import pymysql

# Read JSON file
with open(r'C:\Users\ctlan\OneDrive\desktop\manage␣
    bigdata\assignment\DocumentDB_and_NoSQL\json_award.json',
    ’r’) as f:
    data = json.load(f)

# Connect to MySQL
connection = pymysql.connect(host=’localhost’,
                             user=’root’,
                             password=’@@Ct123456’,
                             database=’hw4_part2_2’,
                             charset=’utf8mb4’)

#table 2
cursor = connection.cursor()

# Create main table if it doesn’t exist
sql_create_table_main = """
CREATE TABLE IF NOT EXISTS Nobel_document (
    index_id INT PRIMARY KEY AUTO_INCREMENT,
    awardYear INT,
    category_en VARCHAR(255),
    category_no VARCHAR(255),
    category_se VARCHAR(255),
    categoryFullName_en VARCHAR(255),
    categoryFullName_no VARCHAR(255),
    categoryFullName_se VARCHAR(255),
    prizeAmount INT,
    prizeAmountAdjusted INT,
    links_rel VARCHAR(255),
    links_href VARCHAR(255),
    links_action VARCHAR(255),
    links_types VARCHAR(255),
    laureate_id VARCHAR(255),
    laureate_name_en VARCHAR(255),
    laureate_portion VARCHAR(255),
    laureate_sortOrder VARCHAR(255),
    laureate_motivation_en VARCHAR(1000),
    laureate_motivation_se VARCHAR(1000),
    laureate_links_rel VARCHAR(255),
    laureate_links_href VARCHAR(255),
    laureate_links_action VARCHAR(255),
    laureate_links_types VARCHAR(255)
);
"""
cursor.execute(sql_create_table_main)
connection.commit()

# Insert JSON data into table
```

```
for item in data:
    awardYear = item[’awardYear’]
    category_en = item[’category’][’en’]
    category_no = item[’category’][’no’]
    category_se = item[’category’][’se’]
    categoryFullName_en = item[’categoryFullName’][’en’]
    categoryFullName_no = item[’categoryFullName’][’no’]
    categoryFullName_se = item[’categoryFullName’][’se’]
    prizeAmount = item[’prizeAmount’]
    prizeAmountAdjusted = item[’prizeAmountAdjusted’]
    links_rel = item[’links’][’rel’]
    links_href = item[’links’][’href’]
    links_action = item[’links’][’action’]
    links_types = item[’links’][’types’]
    if ’laureates’ in item and item[’laureates’]:
        laureate = item[’laureates’][0]
    else:
        laureate = {}

    laureate_id = laureate.get(’id’, None)
    laureate_name_en = laureate.get(’knownName’, {}).get(’en’,
    None)
    laureate_portion = laureate.get(’portion’, None)
    laureate_sortOrder = laureate.get(’sortOrder’, None)
    laureate_motivation_en = laureate.get(’motivation’,
    {}).get(’en’, None)
    laureate_motivation_se = laureate.get(’motivation’,
    {}).get(’se’, None)
    laureate_links_rel = laureate.get(’links’, {}).get(’rel’,
    None)
    laureate_links_href = laureate.get(’links’, {}).get(’href’,
    None)
    laureate_links_action = laureate.get(’links’,
    {}).get(’action’, None)
    laureate_links_types = laureate.get(’links’,
    {}).get(’types’, None)

    sql = """INSERT INTO Nobel_document (awardYear, category_en,
    category_no, category_se, categoryFullName_en,
    categoryFullName_no, categoryFullName_se, prizeAmount,
    prizeAmountAdjusted, links_rel, links_href, links_action,
    links_types, laureate_id, laureate_name_en,
    laureate_portion, laureate_sortOrder,
    laureate_motivation_en, laureate_motivation_se,
    laureate_links_rel, laureate_links_href,
    laureate_links_action, laureate_links_types)
```

```
            VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
    %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s);"""

    cursor.execute(sql, (
    awardYear, category_en, category_no, category_se,
    categoryFullName_en, categoryFullName_no,
    categoryFullName_se,
    prizeAmount, prizeAmountAdjusted, links_rel, links_href,
    links_action, links_types, laureate_id, laureate_name_en,
    laureate_portion, laureate_sortOrder,
    laureate_motivation_en, laureate_motivation_se,
    laureate_links_rel,
    laureate_links_href, laureate_links_action,
    laureate_links_types))

# Commit changes and close connection
connection.commit()
cursor.close()
connection.close()
```

2.2.5 Table2 Query Code – sql

```
literate
use hw4_part2_2;
select
awardYear,
category_en,
laureate_name_en
from nobel_document
where awardYear ='1901';
```

2.3 Table3: Columns

2.3.1 Table3 SnapShot

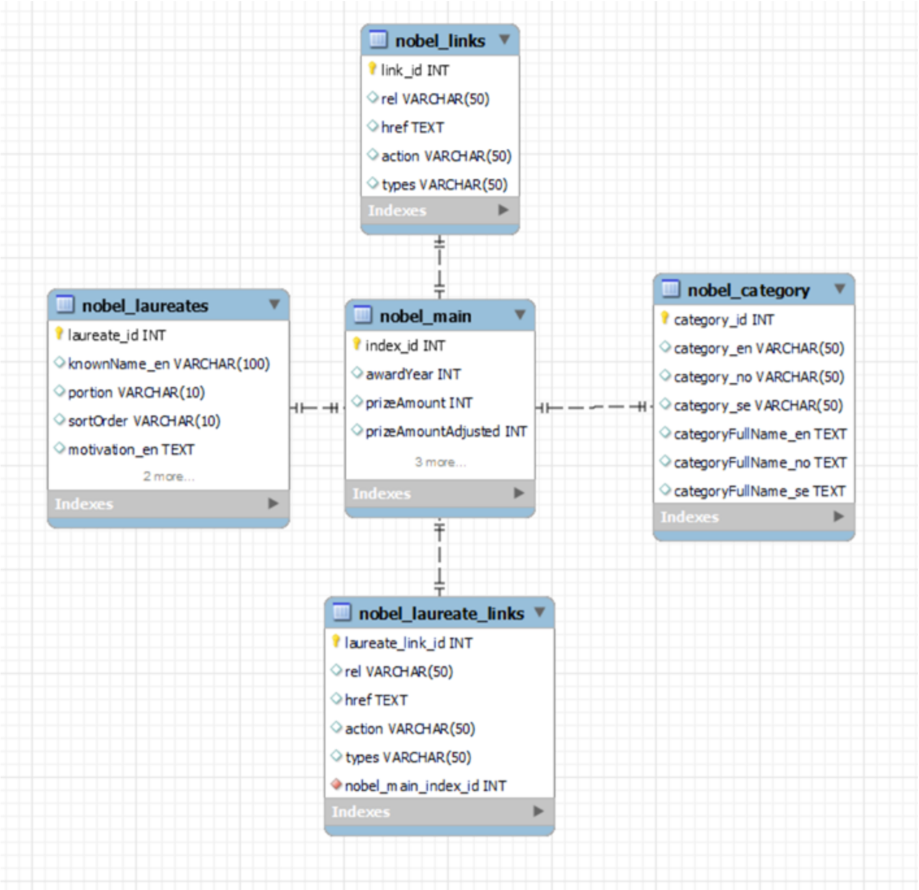
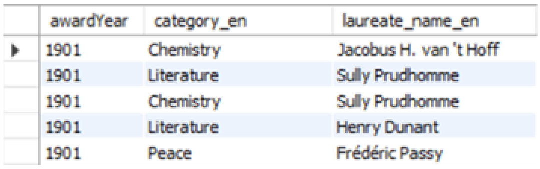


Figure 8: Table3 SnapShot

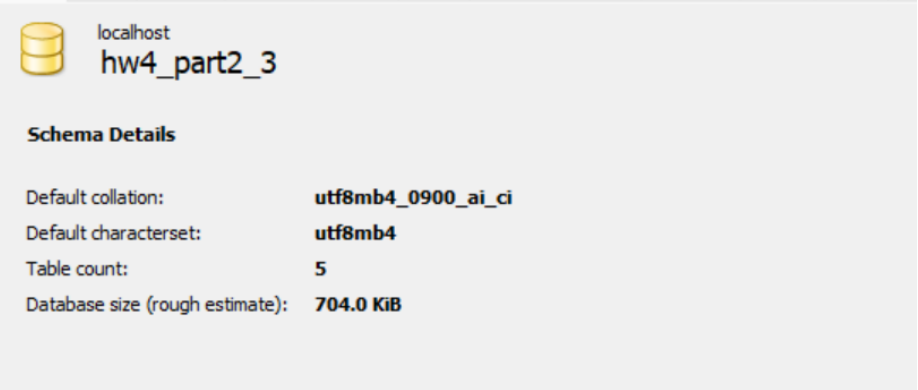
2.3.2 Table3 Query Result



	awardYear	category_en	laureate_name_en
▶	1901	Chemistry	Jacobus H. van 't Hoff
	1901	Literature	Sully Prudhomme
	1901	Chemistry	Sully Prudhomme
	1901	Literature	Henry Dunant
	1901	Peace	Frédéric Passy

Figure 9: Table3 Query Result

2.3.3 Table3 Table Size



localhost

hw4_part2_3

Schema Details

Default collation:	utf8mb4_0900_ai_ci
Default characterset:	utf8mb4
Table count:	5
Database size (rough estimate):	704.0 KiB

Figure 10: Table3 Table Size

2.3.4 Table3 Load Data Code – python

```
literate
import json
import pymysql

# Establish MySQL connection
conn = pymysql.connect(host='localhost', user='root',
    password='@@Ct123456', db='hw4_part2_3')
cursor = conn.cursor()

# Create tables
cursor.execute("""
CREATE TABLE IF NOT EXISTS nobel_main (
    index_id INT PRIMARY KEY AUTO_INCREMENT,
    awardYear INT,
    prizeAmount INT,
    prizeAmountAdjusted INT
);
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS nobel_category (
    category_id INT PRIMARY KEY AUTO_INCREMENT,
    category_en VARCHAR(50),
    category_no VARCHAR(50),
    category_se VARCHAR(50),
    categoryFullName_en TEXT,
    categoryFullName_no TEXT,
    categoryFullName_se TEXT
);
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS nobel_links (
    link_id INT PRIMARY KEY AUTO_INCREMENT,
    rel VARCHAR(50),
    href TEXT,
    action VARCHAR(50),
    types VARCHAR(50)
);
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS nobel_laureates (
    laureate_id INT PRIMARY KEY AUTO_INCREMENT,
    knownName_en VARCHAR(100),
    portion VARCHAR(10),
    sortOrder VARCHAR(10),
    motivation_en TEXT,
    motivation_se TEXT
);
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS nobel_laureate_links (
    laureate_link_id INT PRIMARY KEY AUTO_INCREMENT,
    rel VARCHAR(50),
    href TEXT,
    action VARCHAR(50),
    types VARCHAR(50)
);
""")

# Read JSON file
with open(r’C:\Users\ctlan\OneDrive\desktop\manage␣
    bigdata\assignment\DocumentDB_and_NoSQL\json_award.json’,
    ’r’) as f:
    data = json.load(f)

# Insert data
for item in data:
    cursor.execute("INSERT␣INTO␣nobel_main␣(awardYear,␣
    prizeAmount,␣prizeAmountAdjusted)␣VALUES␣(%s,␣%s,␣%s);",
                   (item['awardYear'], item['prizeAmount'],
    item['prizeAmountAdjusted']))
    main_id = cursor.lastrowid

    category = item['category']
    cursor.execute(
        "INSERT␣INTO␣nobel_category␣(category_en,␣category_no,␣
    category_se,␣categoryFullName_en,␣categoryFullName_no,␣
    categoryFullName_se)␣VALUES␣(%s,␣%s,␣%s,␣%s,␣%s,␣%s);",
        (category['en'], category['no'], category['se'],
    item['categoryFullName']['en'],
    item['categoryFullName']['no'],
         item['categoryFullName']['se']))

    links = item['links']
    cursor.execute("INSERT␣INTO␣nobel_links␣(rel,␣href,␣action,␣
    types)␣VALUES␣(%s,␣%s,␣%s,␣%s);",
                   (links['rel'], links['href'],
    links['action'], links['types']))

    if 'laureates' in item:
        for laureate in item['laureates']:
            knownName_en = laureate['knownName']['en'] if
    'knownName' in laureate and 'en' in laureate[
                'knownName'] else None
            portion = laureate['portion'] if 'portion' in
    laureate else None
            sortOrder = laureate['sortOrder'] if 'sortOrder' in
    laureate else None
            motivation_en = laureate['motivation']['en'] if
    'motivation' in laureate and 'en' in laureate[
                'motivation'] else None
            motivation_se = laureate['motivation']['se'] if
    'motivation' in laureate and 'se' in laureate[
                'motivation'] else None

            cursor.execute(
                "INSERT␣INTO␣nobel_laureates␣(knownName_en,␣
    portion,␣sortOrder,␣motivation_en,␣motivation_se)␣VALUES␣
    (%s,␣%s,␣%s,␣%s,␣%s);",
                (knownName_en, portion, sortOrder,
    motivation_en, motivation_se))

            laureate_links = laureate['links']
            cursor.execute("INSERT␣INTO␣nobel_laureate_links␣
    (rel,␣href,␣action,␣types)␣VALUES␣(%s,␣%s,␣%s,␣%s);",
                           (laureate_links['rel'],
    laureate_links['href'], laureate_links['action'],
                            laureate_links['types']))

    conn.commit()

# Close connection
cursor.close()
conn.close()
```

2.3.5 Table3 Query Code – sql

```
literate
use hw3_part2_3;
select
awardYear,
category_en,
knownName_en laureate_name_en
from
        nobel_category a
inner join
        nobel_laureates b
on a.category_id  = b.laureate_id
inner join
        nobel_main c
on a.category_id  = c.index_id
where c.awardYear ='1901';
```